Probabilistic Models for Robot-Based Object Segmentation

Daniel Beale[a,*], Pejman Iravani[b], Peter Hall[a]

[a]*Department of Computer Science, University of Bath, Bath, BA2 7AY, England*
[b]*Department of Mechanical Engineering, University of Bath, Bath, BA2 7AY, England*

Abstract

This paper introduces a novel probabilistic method for robot based object segmentation. The method integrates knowledge of the robot's motion to determine the shape and location of objects. This allows a robot with no prior knowledge of its workspace to isolate objects against their surroundings by moving them and observing their visual feedback. The main contribution of the paper is to improve upon current methods by allowing object segmentation in changing environments and moving backgrounds. The approach allows optimal values for the algorithm parameters to be estimated. Empirical studies against alternatives demonstrate clear improvements in both planar and three dimensional motion.

Keywords: Active Segmentation, Pattern Recognition, Robot Vision, Sensor Fusion, Robot Learning, Probabilistic Modelling

1. Introduction

For a robot to behave autonomously in a free environment it must be able to segment, localise and identify objects that it has not previously encountered. Localisation means identifying the place in the visual field where the object lies, segmentation means specifying the object's boundary, and identification means to recognise or classify it. These are large questions, and they are intimately linked so that a full discussion is well beyond the bounds of this paper. Instead, we take the view that segmentation and localisation are precursors to identification. This paper shows how our probabilistic motion model can be used by an autonomous robot to segment and localise an object in a video, even when the background is cluttered and in motion.

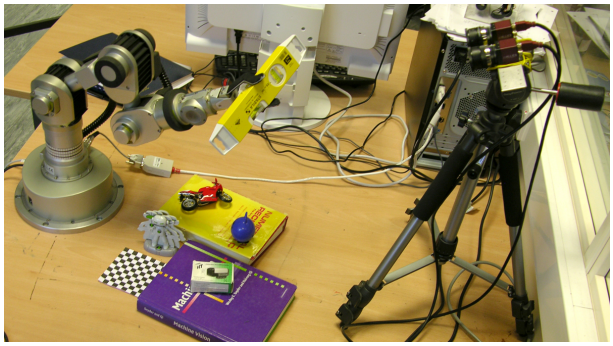

Figure 1: The experimental set up

Localisation in Computer Vision could mean drawing a bounding box around the object, for example face detectors do not have to decide the boundary of a face but may operate on feature (eye, mouth, nose) detection and infer a bounding box from them. Segmentation in Computer Vision means breaking an image or video into semantically meaningful parts. Segmentation is a vast field but can be broadly decomposed into two parts. Low-level methods operate using data drawn from images alone, typically assuming coherence properties of some kind (colour, texture, *etc.*). Low-level approaches are applicable to a wide class of inputs, but the problem of isolating an object from the background remains. High-level methods address this problem using pre-learned object models, these are typically not suitable for autonomous robots in free environments because they restrict localisation to objects for which models already exist. However, if a robot can learn new high level models bottom-up (that is, by inference from low-level data) then that robot is able to gain all of the advantages high-level offers: compact representation, ability to plan, ability to recognise and so on.

We define *object segmentation* as a short hand for localising and segmenting an object. This paper is motivated by the following idea: when applied in a robotics scenario, low-level algorithms can be enhanced by integrating information from the robotic system to improve the quality of object segmentation or to eliminate the need for high-level assumptions. This is because touching objects provides enough low-level information to separate them from the general background. Once an object has been isolated it is, in principle, possible for a robot to learn much, including visual appearance, physical properties and so on. We will not address such issues in this paper; we limit our contribution to object segmentation.

Object segmentation is important in the robotic community as it will allow robots to manipulate objects in cluttered backgrounds. For example, vision can be used to direct a robot arm towards a target (Weiss et al., 1987; Chaumette and Hutchinson,

*Corresponding author
Email addresses: d.beale@bath.ac.uk (Daniel Beale), p.iravani@bath.ac.uk (Pejman Iravani), pmh@cs.bath.ac.uk (Peter Hall)

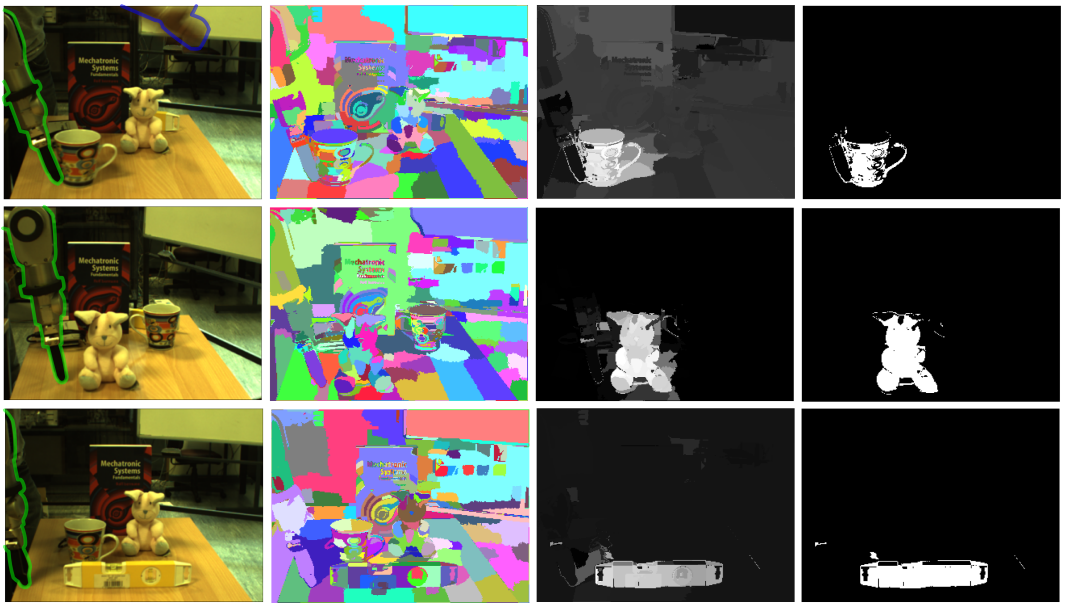

Figure 2: Results from the *Motion-Likelihood* method. From left to right: (1) A frame from the video. (2) The corresponding low-level segmentation. (3) The resulting probability map over each segment. (4) The thresholded probability map corresponding to the object of interest. In the left column the robot arm has been manually outlined in green, and a moving human hand in blue; these outlines are for the convenience of readers only — they play no part in our algorithm.

2006, 2007) or to direct a mobile robot towards potential landmarks (Espinace et al., 2008), and recognise objects in novel scenes (Welke et al., 2010).

In this paper we use robot motion data to extend segmentation algorithms to localise objects. The basic idea is to segment a video into pieces, and correlate any motion of those pieces with the motion of the robot arm. The two main advantages of the presented method over state of the art segmentation/localisation algorithms are:

- The background is not assumed to be static, thus improving the algorithm's stability in changing environments
- The localisation is probabilistic, giving a degree of certainty that any part of image belongs to the object

In addition, the probabilistic model we provide is versatile, this paper shows examples based on motion in the plane, and examples of three dimensional motion under a homography. Moreover, the testing regime we use automatically provides estimates for the optimal of parameters that control the object isolation algorithm.

The ability to localise against a moving background is important. Consider that for a robot in a free environment there is no guarantee the background will be static; also a robot may move its head — if only through vibrations from the motors. This means localisation algorithms that rely on background subtraction, *e.g.* (Fitzpatrick and Metta, 2003) will not work. Our approach solves this problem.

Figure 1 illustrates the experimental test bed which comprises of two cameras and an articulated arm. Figure 2 illustrates some of the stages and the final output of the method. The paper first defines the probabilistic motion model and its application, then implements the method and presents the experimental results. The full process, named *Motion-Likelihood* (M-L), is shown to outperform state of the art alternatives.

2. Previous Work

The prime interest in this paper is object segmentation. Let us define this as any process that identifies the set of pixels in a video that belong to the object. Localisation is related to segmentation in that the object must be segmented from its background. However, 'segmentation' normally refers to breaking a video (or image) into regions that are at best putative objects. Problems such as occlusion can divide an object into two or more regions or obscure part of it; shadow is problematic for similar reasons; the visual clutter of other objects can mean several objects are merged into one region. Even so, we use a state of the art segmentation algorithm to generate regions, and show how a robot that pushes objects can robustly localise individual objects, bottom-up.

Approaches to object segmentation can be categorised into three different classes. (i) Vision-only object segmentation using priors, (ii) low-level segmentation combined with motion information and (iii) robot induced motion for segmentation.

The computer vision community provides large amounts of research in the field of segmentation. Breakthroughs have been made in high-level static image segmentation using prior knowledge about objects in the scene. For example (Arbelaez et al., 2009) uses human prior knowledge, (Yin and Collins, 2009) use shape priors, and (Stein et al., 2007; Huang et al.,

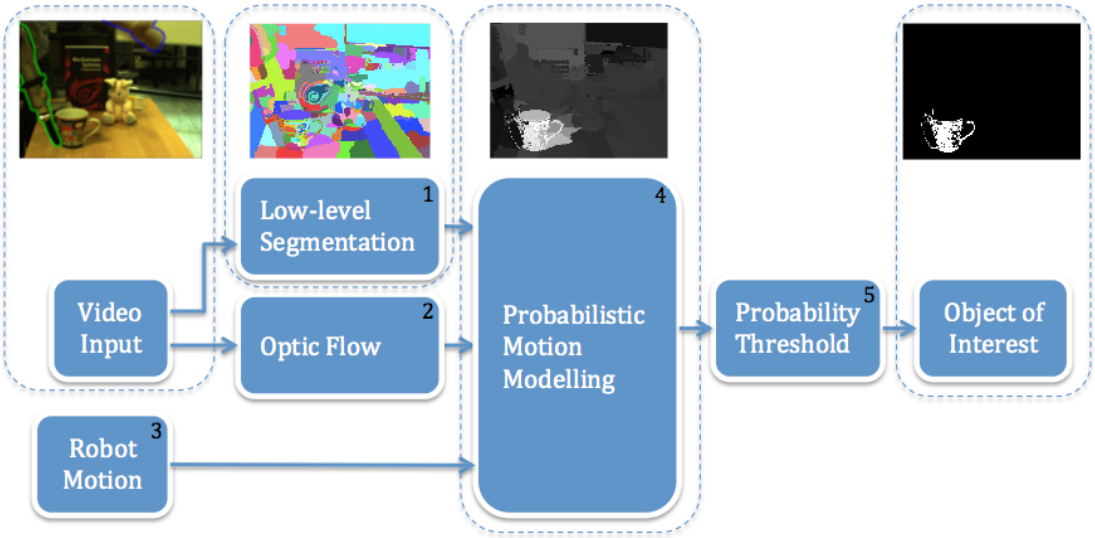

Figure 3: A diagram showing the full object segmentation process. The method takes input from a camera and the robot arm, the images are processed using low-level segmentation and optical flow. The probability that each segment matches the arm motion is calculated over a period of time. The resulting probability map is then threshold to obtain a final object-background segmentation.

2009) use motion priors to improve the quality and accuracy of the segmentation. Using prior knowledge about objects is beneficial but is not always available. For a robot to learn about objects in the world autonomously it needs to be able to build its own representations and priors, allowing it to adapt to its environment. Therefore, using low prior knowledge for learning is better motivation towards independent, adaptable robots.

Low-level segmentation algorithms use only pixel information to split images (Comaniciu and Meer, 2002; Felzenszwalb and Huttenlocher, 2004) and videos (Paris, 2008; Brox and Malik, 2010; Vazquez-Reina et al., 2010) in to meaningful regions, but these are not object-dependent. The work of (Brox and Malik, 2010; Ross, 2000) performs general object-level segmentation on videos by analysing optical flow trajectories. These algorithms provide a useful basis for a robot to decide which parts of a video are moving together, and thus defining objects. Despite this, the problem of finding the object that is being purposely moved by the robot remains unanswered.

Using robot motion to segment objects was introduced in an experiment designed to show how robots can learn in a similar way to humans (Metta and Fitzpatrick, 2003). In the experiment, the robot begins with no information of the world, but through interacting with it, segmentations and descriptions of objects can be found (Fitzpatrick and Metta, 2003). These results have recently been extended to learning about multiple objects in unstructured environments (Kenney et al., 2009), allowing a robot to isolate objects in the scene and independently track them. The quality of segmentation is improved if the robot is allowed to touch the object multiple times. Assuming that the robot is holding the object can also provide a powerful cue for segmentation (Arsenio et al., 2003; Ude et al., 2008), more recently extended by (Welke et al., 2010), combining proprioceptive information with background subtraction and visual disparity. These methods require that a human places the object in the grasp of the robot, making the methods 'supervised' approaches to learning. Using a human supervisor to identify the object in this way provides prior information that will not always be available to an autonomous robot, the method presented in this paper provides the foundations for an approach that is unsupervised, without the need for human interaction. A common characteristic of the above methods is that they all make the assumption that a single object in the environment is moving at any given time (*i.e.* in (Kenney et al., 2009) multiple objects can be segmented only if one moves at any time). This assumption leads to the algorithms becoming unstable if the background changes at the same time the objects move.

The method presented in this paper is different from the above methods as: (i) it assumes no prior knowledge about the objects, (ii) a correlation is identified between object motion and the robot's arm motion allowing the robot to segment only the object it has moved, and (iii) the background is not assumed to be static. Although some of these characteristics can be found in previous work, none address all of them together. The work presented in this paper is compared experimentally to (Fitzpatrick and Metta, 2003) showing an improved quality of segmentation.

3. Probabilistic Motion Models

The method introduced in this paper allows a robot to manipulate an object in front of it (*e.g.* push or grasp), and use the visual information to localise the object against its background. This is done by a probabilistic process that correlates robot and image motion. The underlying intuition behind the method is that objects move in a coherent and predictable manner.

Table 1: Notation

Q	$\mathbb{R}^n \to \mathbb{R}^n$ Motion transformation function
Φ	Probabilistic motion model
ϱ	$\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ Error measurement function
$p(\varepsilon\|Q,\Omega)$	$\mathbb{R} \to [0,1]$ Error probability density function
Ω	Prior data relating to error distribution
x_i	Robot end effector position in camera coordinates at frame i
ε	Error value
$\mathcal{S}$	Set of segments in an image
S_j	the jth segment in an image
$\mathbf{g}$	$\mathbb{R}^5 \to \mathbb{R}^3$ Forward kinematics function
P	$\mathbb{RP}^3 \to \mathbb{RP}^2$ Camera calibration matrix
$\mathbf{h}$	$\mathbb{R}^5 \to \mathbb{R}^2$ Function mapping robot joint space position into camera coordinates
I	Identity matrix
Σ	Covariance matrix
$N()$	Normal distribution
T	Total set of frames in one experiment
$\hat{T}$	$\hat{T} \in T$ Arbitrary subset of frames

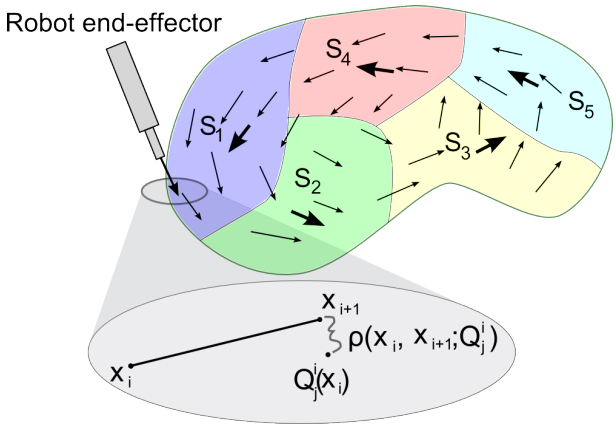

Figure 4: A diagram showing how to assess arm motion membership to a model. The arrows in the object denote the optical flow field. Each S_j represents a different segment, each with their own motion model (dark arrow). The large ellipse represents a magnification of a motion model being tested against the motion of the end-effector.

The correlation between the motion on the video and the robot arm is evaluated using a statistical motion model. A main component of the motion model is the motion transformation function. This function is used to represent how things can move in the environment. For example a rigid body motion transformation (*i.e.* objects that cannot be deformed) implies that the pixels within the object must 'move together'. Other motion transformation functions, such as elastic or fluidic could be used. The motion transformation function is used to calculate the error between the observed motion in the image and the expected one for a given arm motion. In other words, how things should move given that they are rigid and they've been pushed in a particular manner.

The method presented here comprises five steps:

1. Low-level image segmentation
2. Computing the motion in the image using dense optic flow
3. Mapping end-effector motion into camera coordinates
4. Estimating the arm-segment motion correlation
5. Thresholding the estimation to define the object of interest

In step 1, the low-level segmentation of (Paris, 2008) is used, this takes a video stream and segments it according to the colour information in each frame. This gives the robot knowledge of where the segments are, and tracks them throughout the video. In principle, any other segmentation algorithm could be used, (Paris, 2008) was selected for the quality and speed of computation. Step 2 is calculated using optical flow. A fast implementation from OpenCV (Bouguet, 1999) is used. In Step 3 the robot joint angles are converted into 2D camera coordinates. Step 4, estimates the correlation of segment motions with that of the robot arm. The result is a probability that any segment belongs to the manipulated object. Finally, Step 5 chooses the segments with highest probability and defines the object of interest. Figure 3 illustrates a diagram of the method. The blocks illustrate the previous steps and the images some of their outputs.

The process to calculate motion models and correlate them with the arm motion is detailed in the rest of this section. This process is inspired by the work of (Torr and Zisserman, 1998; Torr, 1998) for general motion segmentation.

3.1. Motion Model Definition

The objective of modelling motion is to use the model information to segment an object. By correlating the motion of the arm to that of a region in the image, objects being manipulated can be discerned from the background. This section defines the novel probabilistic motion model.

In general terms, let us define a probabilistic motion model as a 3-tuple, $\Phi = (Q, \varrho, p(\cdot|Q,\Omega))$, where each element is defined in Table 1.

The motion transformation function, Q^i_j, is calculated as a function of the optic flow of each pixel with the jth segment in the ith frame. For example, Q could be simply calculated as the average optic flow within a segment (as in Section 4.2.1) or as a Homography transformation (as in Section 4.2.2). Then, the position for any arbitrary point in the image, $\mathbf{a}$, undergoing this transformation can be calculated as $Q^i_j(\mathbf{a})$. The same transformation can be applied to the initial arm position $\mathbf{x}_i$, resulting on the expected arm position if this was moving like segment j, $Q^i_j(\mathbf{x}_i)$. The transformed arm position can then be compared to the real arm position at frame $i+1$, $\mathbf{x}_{i+1}$. The error between these two indicates how correlated they are. This error could be defined, for example, by the distance $\varrho(\mathbf{x}_i, \mathbf{x}_{i+1}; Q) = |Q(\mathbf{x}_i) - \mathbf{x}_{i+1}|^2$. The probability distribution of the error, $p(\varepsilon|Q,\Omega)$, is then used to assess the membership between the robot and the segment's motion as detailed in the next section. Figure 4 illustrates an example where motion transformation functions are defined as the average segment motion.

3.2. Assessing Segment Membership through Motion

This section details how the motion of a segment is correlated to that of the robot arm using the probabilistic motion model. The model estimates the probability of observing a particular error magnitude between the end-effector and the segment's estimated motion model. The probability that a particular motion model, Q, explains the observed error is needed, and so an appeal is made to Bayes' theorem to obtain

$$p(Q|\varepsilon, \Omega) \propto p(\varepsilon|Q, \Omega)p(Q|\Omega). \tag{1}$$

The posterior conveniently factorises into the likelihood of the error and a prior on the transformation function. The prior is used when extra information is available about the object regarding its shape, colour or location. For example, this could allow the use of previously learned object models to influence the decision about which segments belong to the object being moved. If no prior is known, then Ω is regarded as the empty set.

4. Implementation of the Method

The method described in this section allows a robotic arm to poke and grasp objects in order to discriminate them from their background. The paper ignores many of the details regarding robot control and trajectory generation as these are not relevant to the contribution. This section details the steps required to practically implement the method.

The motion of regions in the video are extracted and correlated with the arm movement as explained in Section 3.2. This gives a probability that each segment belongs to the object, which is then thresholded to extract an object-level segmentation. A prior can optionally be used to give extra information about the object, for instance its colour distribution, or shape.

4.1. Robot and Camera Calibration

To effectively use information from multiple sensors it is important to ensure that all measurements are in a common coordinate frame. In the context of robotic manipulation with a computer vision system this is generally known as *robot calibration*. The robot and camera system are calibrated as follows:

1. Joint position information is transformed to 3D task space using forward kinematics. This is generally well known in the literature (Spong and Vidyasagar, 2007) and can be computed using standard trigonometry. This function is denoted $\mathbf{g} : \mathbb{R}^5 \rightarrow \mathbb{R}^3$.
2. The resulting 3D points in the task space are mapped to the 2D camera coordinate frame by positioning the robot to a wide set of points within its reach. The position of the end-effector is recorded, and a homogeneous projection matrix $P : \mathbb{RP}^3 \rightarrow \mathbb{RP}^2$ is computed numerically, mapping the 3D robot task space into the camera frame. This is known in the literature as camera calibration (Hartley and Zisserman, 2000).

The resulting function, Equation 2, maps the joint positions into the camera coordinates.

$$\mathbf{h} : \mathbb{R}^5 \rightarrow \mathbb{R}^2 \tag{2}$$

$$\mathbf{h}(\mathbf{x}) = \pi(P(\mathbf{g}(\mathbf{x}))) \tag{3}$$

where $\mathbf{h}$ is the function that maps joint information into the camera frame and $\mathbf{g}$ is the robot's forward kinematic function.

The function $\pi : \mathbb{RP}^2 \rightarrow \mathbb{R}^2$ represents the embedding of homogeneous coordinates into the camera coordinate. This is a division by and removal of the last element of the input.

4.2. Motion Transformation Function

The theory laid out in Section 3 gives an abstract way to define probabilistic motion models. In this section the definition is applied to the problem of object-level segmentation using a robot arm.

A video stream is segmented using the algorithm of (Paris, 2008). The algorithm tracks each segment S_j throughout the video stream. A motion model is then assigned to each segment, j, so that the probability of a segment correlating with the arm movement can be found.

4.2.1. Translation transformation function

Assuming that segments can only undergo a translation motion in the camera space, the model, $\Phi^i_j = (Q^i_j, \varrho, p(\cdot|Q^i_j, \Omega_i))$, can be instantiated as follows:

$$Q^i_j(\mathbf{a}) = \mathbf{a} + \mathbf{b}^i_j \tag{4}$$

$$\varrho(\mathbf{x}_i, \mathbf{x}_{i+1}; Q^i_j) = |Q^i_j(\mathbf{x}_i) - \mathbf{x}_{i+1}|^2 \tag{5}$$

Where Q^i_j is given by a single translation, i is the frame index, $\mathbf{a}$ is a variable position vector, and $\mathbf{b}^i_j$ the average segment motion. This average motion is calculated using dense optical flow (Bouguet, 1999). As previously, $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are the position of the arm at frames i and $i+1$ respectively. The error distribution $p(\varepsilon|Q^i_S, \Omega_i)$ is calculated as a Gaussian with a mean of $\mathbf{x}_i - \mathbf{x}_{i+1}$ and covariance Σ. The norm given above is the 'L2 norm' $|\mathbf{x}| = \sqrt{\sum_{k=0}^{K} x_k^2}$.

4.2.2. Homography transformation function

It is possible to extend the practical implementation in to three dimensions. Allowing the robot to segment objects which move non-linearly.

The tensor which models rigid body motion in 3D from observations in a single camera is known as the fundamental matrix (Hartley and Zisserman, 2000). Points which match across frames $\mathbf{x}_i, \mathbf{x}_{i+1} \in \mathbb{RP}^2$ are related as follows:

$$\mathbf{x}_i^T F \mathbf{x}_{i+1} = 0 \tag{6}$$

This relation can be used as a motion model in the formulation described in 3.1. Setting $\varrho(\mathbf{x}_i, \mathbf{x}_{i+1}) = \mathbf{x}_i^T F \mathbf{x}_{i+1}$, the density function $p(\varepsilon|Q, \Omega)$ can then be accurately approximated as χ^2. The

transformation function Q can't be directly computed using this approach, but the implementation only requires the error density function to compute the object segmentation.

If an extra assumption is made that the points inside a segment are planar, then a homography $H \in \mathbb{R}^{3\times3}$ can be used for the motion model:

$$H^i_j \mathbf{x}_i = \mathbf{x}_{i+1} \tag{7}$$

$$Q^i_j(\mathbf{a}) = H^i_j \mathbf{a} \tag{8}$$

$$\varrho(\mathbf{x}_i, \mathbf{x}_{i+1}; Q^i_j) = |Q^i_j(\mathbf{x}_i) - \mathbf{x}_{i+1}|^2 \tag{9}$$

Both equations 6 and 8 represent a body moving rigidly in 3D. These models can be used to extract a full 3D model of the object using a single camera. Alternatively, if dense 3D optical flow can be obtained from a pair of cameras then a rigid body transformation function can be used for $Q : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. The probabilities and low-level segmentation algorithm remains the same.

The fundamental matrix can be computed from a pair of homographies (Pellejero et al., 2004). Combining this approach with the object segmentation algorithm detailed here, and dense monocular SLAM (Newcome and Davidson, 2010), the motion models could be used to compute a reconstruction of the object from the video. This is out of scope for this paper, but would be an interesting extension for future work.

4.3. Error Probability Density Function

Given the model parameters, the error probability density function can be calculated as follows:

$$p(\varepsilon|Q^i_j, \Omega_i) = N(\mathbf{x}_i + \mathbf{b}^i_j - \mathbf{x}_{i+1}|0, \Sigma) \tag{10}$$

$$= N(\mathbf{x}_i - \mathbf{x}_{i+1}|\mathbf{b}^i_j, \Sigma) \tag{11}$$

$$p(\varepsilon|Q^i_j, \Omega_i) = N(H^i_j \mathbf{x}_i - \mathbf{x}_{i+1}|0, \Sigma) \tag{12}$$

Where Equations 11 and 12 represent the error probability density function for the translation and homography transformation functions respectively.

The covariance $\Sigma = \lambda I$ is chosen with parameter $\lambda \in \mathbb{R}$. In practise, choosing any value for λ between 10 and 50 does not affect the results significantly.

This definition of the error density assigns a high probability to segments that belong to the robot arm. In order to separate arm segments from those of the object, the next section defines a new probability density function that eliminates highly probable arm segments.

4.4. Removing Arm Segments from the Object

Correlating all movement in the video with that of the arm results in the arm itself being segmented as a part of the object of interest. To prevent this from occurring, the arm segments must be eliminated from the video.

The standard way to remove the arm segments would be to use an accurate 3D model of the arm, and project it on to the cameras (Welke et al., 2010). This paper presents a probabilistic approach which is a simpler alternative, as calculating the kinematics and calibrating a full CAD model of the robot is time consuming, and assumes that an accurate model of the robot is available. Further, the method used here could be combined with model-based arm removal for improved accuracy and robustness.

At each frame, the position of the arm's end effector is recorded together with a binary variable indicating whether or not an object is being touched; a simple sensor is sufficient to record this. This binary variable is needed so that the robot arm is not segmented as the foreground as follows.

Any motion that correlates with the arm movement before an object is touched is likely to belong to the arm. Any segments which correlate to the arm motion after the object is touched are both part of the object and the arm. This information is used to subtract the arm from the segmentation. Thus, the full error distribution is:

$$p(\varepsilon|Q^i_j, \Omega_i) = \begin{cases} \frac{1}{\kappa^i_j}(1 - p(\varepsilon|Q^i_j, \Omega_i)) : i \in \hat{T} \\ p(\varepsilon|Q^i_j, \Omega_i) : i \notin T' \end{cases} \tag{13}$$

with,

$$\kappa^i_j = \int_{\mathbf{x}_i, \mathbf{x}_{i+1} \in D} 1 - p(\varrho(\mathbf{x}_i, \mathbf{x}_{i+1}; Q^i_j)|Q^i_j, \Omega_i) \tag{14}$$

Where $D = [-N, N] \times [-M, M]$ is the set of all possible values for the optical flow, M and N are the height and width of the image respectively, and $\hat{T}$ is the subset of frames in which the object is being pushed. In practise κ^i_s can be any constant since each i is independent. In this formulation $\Omega_i = (i, T, \Sigma)$. In Figure 5 it can be seen that the robot arm segments (left side of the images) are darker (lower probability) in comparison to the other segments in the image.

4.5. Integrating over Multiple Frames

So far we have only considered the probability of a segment in a single frame. It remains to calculate the distribution for a set of frames.

Using Equation 1 and substituting Ω by $\Omega^i_j = (i, T, \Sigma)$ and Q by Q^i_j allows prior knowledge to be incorporated at each frame i. Nuisance variable i can be eliminated by marginalisation over the variable $i \in T$. This step can be realised at any point in time, for a set $\hat{T} \subset T$:

$$p(S_j|\varepsilon, T', \Sigma) = \sum_{i \in \hat{T}} p(S_j|\varepsilon, i, T, \Sigma) p(i|\varepsilon, T, \Sigma) \tag{15}$$

$$= \frac{1}{|\hat{T}|} \sum_{i \in \hat{T}} p(Q^i_j|\varepsilon, \Omega_i) \tag{16}$$

The final step follows from assuming that our confidence in the time is uniformly distributed over the whole window $\hat{T}$, and that the probability of a segment S_j given frame i is equal to that of its motion model Q^i_j.

This results in a probability map for each segment throughout the video. The quality of the object segmentation improves as

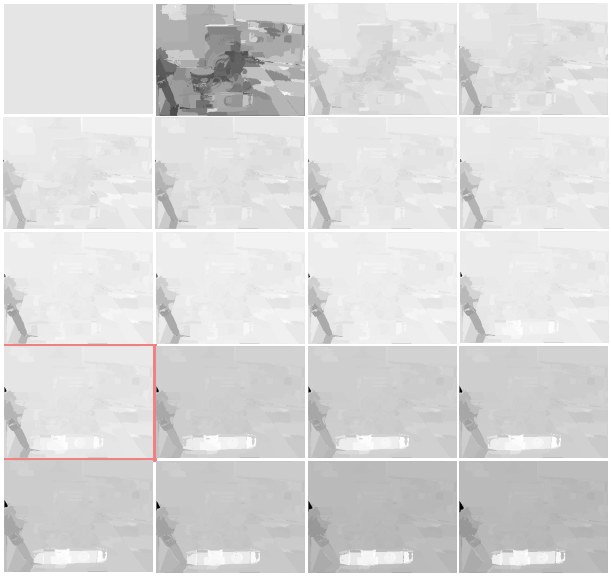

Figure 5: The probability map of a spirit level being pushed, evaluated at a different set of times throughout the video. The top-left image is the map for the first frame and the bottom-right represents the final frame. A red boundary shows the first frame that the object was touched. The spirit level becomes more probable (brighter) with each successive frame.

this probability is integrated. Figure 5 shows the probability map evaluated for a range of different $\hat{T}$. The left most image shows the distribution before the object has been touched, and on the right after the integration over 21 frames. The centre image is the distribution at the time the robot touches the object.

Thresholding the probability map to obtain a segmentation throws away a lot of information. In practise a map is stored as a measure of our confidence in the whereabouts of the object. The robot could then go back and touch areas of the scene that it is uncertain about, transforming the process in to interactive segmentation (Kenney et al., 2009).

5. Results - Linear Translation Arm Motions

The method is tested using a single camera and a Katana 5 DOF robot. A total of 61 videos, of the robot pushing 5 different objects were taken (toy rabbit, mug, spirit level, bottle and book). In this set of videos, the robot moves linearly and parallel to the table that contain the objects (see Figure 1 for the experimental set up). The videos are of a cluttered scene, some with moving backgrounds. The translation transformation function defined in Section 4.2.1 is used here.

The frame in which the objects are touched for first time are hand segmented to give a ground truth segmentation to evaluate the results. This dataset can be obtained by emailing the authors.

5.1. Comparing to current state of the art

The method is compared experimentally to (Fitzpatrick, 2003) since they use robot motion to segment objects. Although (Kenney et al., 2009) provides a more recent version, the underling segmentation algorithm is the equivalent to that presented in (Fitzpatrick, 2003). In both approaches, background subtraction is used to determine the temporal location of the touch event. The frame directly preceding the touch is used as a model for the arm. The Graph Cut algorithm (GraphCuts) (Boykov and Kolmogorov, 2004) is then used to build a hull around these points. The frame in which the object is touched for the first time is applied to GraphCuts, with the pixels that belong to the arm model classified as background.

Some results of our algorithm, Motion-Likelihood (M-L), are shown in Figure 2 showing the full process used to obtain the object-level segmentation. The far left column shows the initial videos of the toy rabbit, cup and spirit level which are being poked by the robot; a low-level segmentation is obtained using (Paris, 2008) shown in the second column; the probability map is given in column three; and the result of thresholding the map is displayed on the far right. It is worth noting here that Fitzpatrick's method can only be computed on the frames nearest to the touch event, and so M-L has the advantage that more information can be used for discriminating between background and object. To make the test as fair as possible, the same frame was chosen for the comparison, although a segmentation exists for the whole video. This verifies that M-L provides a good quality of segmentation compared to a current state of the art technique in a single frame.

Both M-L and (Fitzpatrick, 2003) rely on a free parameter. In (Fitzpatrick, 2003) the authors apply a threshold to the subtracted image. M-L applies a threshold to the probability map. Reasons for choosing this parameter may vary according to the hardware used or visual statistics of the scene. Following the work of (Martin et al., 2004), the segmentation's *Precision* and *Recall* are calculated for every value of the free parameter against the ground truth. These graphs will indicate the optimal parameter values for both segmentation algorithms and compare their performance. Precision (Pr) and Recall (Re) are defined as follows:

$$Pr = \frac{t_p}{t_p + f_p} \tag{17}$$

$$Re = \frac{t_p}{t_p + f_n} \tag{18}$$

Where t_p is the of number true-positives, the pixels that are correct and verified by the ground truth; f_p is the number of false-positives, pixels which are not part of the ground truth but have been detected as being part of the object ; and f_n are number of false-negatives, pixels which are in the ground truth but are not found to be part of the object. If the true object is completely covered by the segmentation then there will be no false-negatives and the recall will be 1, on the other hand, if the object covers the segmentation then there will be no false-positives and the precision will be 1. The segmentation is perfect if and only if both Precision and Recall have a value of 1.

The results of using Precision-Recall (P-R) for every parameter of M-L and (Fitzpatrick, 2003) are shown in Figure 6 for

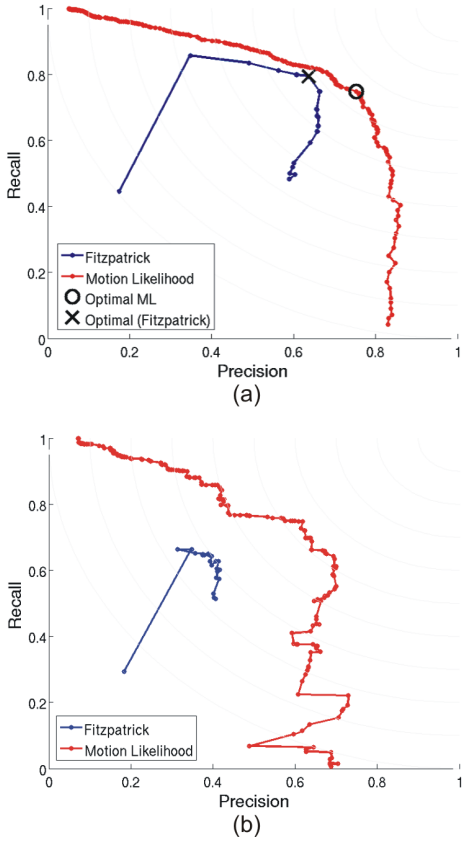

Figure 6: P-R graphs for M-L and (Fitzpatrick, 2003). With static (a) and changing (b) backgrounds.

the five different objects. Figure 6(a) is the P-R for videos with static background and Figure 6(b) with a changing background. The points on the line represent different values of the parameter. The segmentation is of a better quality if it is closer to the top right hand side of the graph (1,1). Both of the graphs show that M-L has a superior segmentation performance to that of Fitzpatrick. Figure 6(b) shows the P-R graph for all of the videos with a changing background. In this test M-L performs significantly better.

Low valued thresholds result in most of the background being considered as part of the object, thus a high recall. M-L exhibits this relationship, with recall decreasing as the threshold increases. The P-R results for (Fitzpatrick, 2003) show a graph with a more unusual shape at a low threshold. This was further investigated and it was found that when the subtraction method used by (Fitzpatrick, 2003) is thresholded at a low value the result is a noisy binary image with foreground distributed across the whole image. Proceeding to use GraphCuts on the image acts as a filter removing a lot of the background noise but inaccurately segmenting the object. As the threshold is increased, the amount of noise reduces, significantly improving the segmentation and the result from GraphCuts.

Although testing the algorithm across each parameter gives good intuition of the performance, a single parameter for the threshold needs to be calculated to verify the performance for practical use.

To determine the quality of the segmentation at a single threshold we introduce a measure of the distance between foreground/background segmentations and ground truth. Specifically we use the average L2 distance between two binary images:

$$\rho_{L2}(Im_1, Im_2) = \frac{1}{NK}\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{K}|Im_1(i,j) - Im_2(i,j)|^2} \quad (19)$$

and the Tanimoto distance, given by,

$$\rho_T(Im_1, Im_2) = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{K}|Im_1(i,j) \wedge Im_2(i,j)|}{\sum_{i=1}^{N}\sum_{j=1}^{K}|Im_1(i,j) \vee Im_2(i,j)|}. \quad (20)$$

The threshold for each algorithm is chosen to be the optimal value in terms of the Precision and Recall. For both M-L and (Fitzpatrick, 2003) the point on the P-R curve for all of the objects that is closest to the point $(1, 1)$ is chosen. This optimal value is shown in Figure 6(a), and takes on a value of 75% of the maximum likelihood for M-L and a threshold of 5 for (Fitzpatrick, 2003). These values are used to segment each video in the dataset for comparison. The measure in Equation 19 is used to find the distance between the object segmentations and ground truth, giving the average distance between two segmentations.

This is done for all of the videos using Fitzpatrick's method, M-L and M-L with the Graph Cut algorithm (ML+GraphCuts). Using GraphCuts draws a hull around the segmented object in order to fill in any details which are missing in the original segmentation. Experimenting with this algorithm determines how well the M-L algorithm performs, if GraphCuts makes a big difference to the result then the M-L performance is poor for details on the object.

Figure 7 shows the results of this experiment. In Figure 7(a), for the book, spirit level and bottle, M-L provides a better quality of segmentation. The low standard deviation, implies it is more stable across a range of different environments. Videos in the dataset include moving backgrounds, camera shake and change in lighting condition. The rabbit performed roughly the same, and the cup object performs slightly worse because of a failure in the low-level segmentation algorithm of (Paris, 2008), due to the complicated pattern on it.

The results were further divided into cases in which the video has a static background and ones with a changing background. Figure 7(b) shows that both methods become a little more unreliable when the background is changing, M-L represents an improvement over Fitzpatrick however. Using M-L with the GraphCuts does not change the performance significantly. In the case of the book the variance has increased, meaning that the quality of segmentation is more unstable across the whole test set. GraphCuts can be seen as a way to patch the the algorithm if it performs badly, but M-L performs well enough without the need for anything else.

Figure 7 visually indicate that our method offers an improvement over Fitzpatrick, the Kolmogorov-Smirnov (KS) test provided a quantitative indicator. In our context, this can be interpreted as the percentage chance that our method outperforms

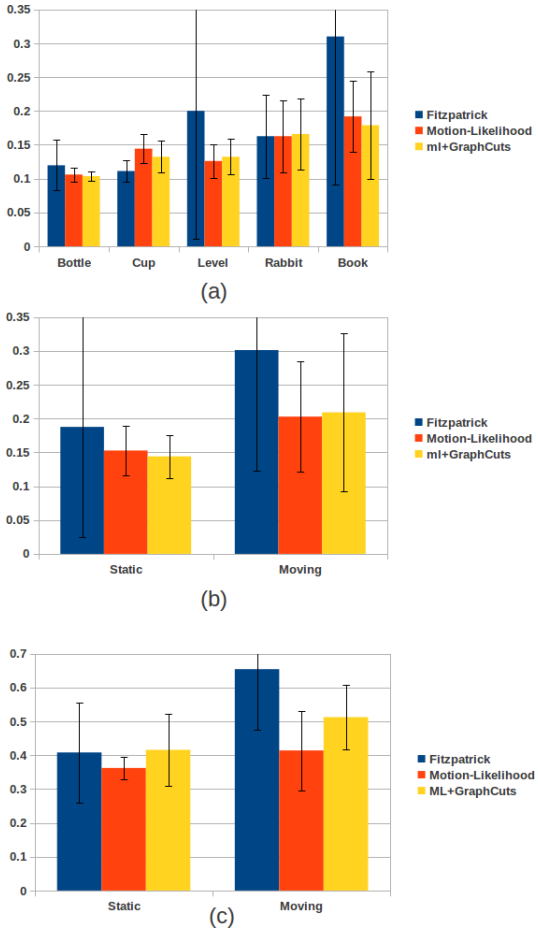

Figure 7: Bar chart comparing segmentation algorithms for both changing and static backgrounds. Each colour represents a different method, the error bars represent the standard deviation. Organised by different objects (a) and backgrounds (b), measured by the L2 distance defined in Equation 19. The bar chart in (c) represents the errors as measured by the Tanimoto distance in Equation 20

Fitzpatricks, given a random scene. In the case of static backgrounds the KS-test gives results of 60% and 45% for ML and graph cuts. For moving backgrounds we get 48% and 95%. The L2 distance, is known to be a poor measure of visual quality; the Tanimoto distance, as defined in Equation 20, is thought to be more reliable for binary images. Figure 7(c) graphically displays the results. The KS results for the Tanimoto distance are as follows, 46% and 61% for ML and graph cuts in static scenes, and 81% for both ML and graph cuts for moving backgrounds. Presenting qualitative results help one to interpret these numbers.

Figure 8 shows a sample of the results for videos with a static 8(a) and changing 8(b) backgrounds. Both methods perform reasonably well when the background is static, although using subtraction poses problems in some of the videos. For example, in the centre column of Figure 8(a). The constant uniform yellow colour of the spirit level means that image differencing does not detect a change when the object is moved between frames.

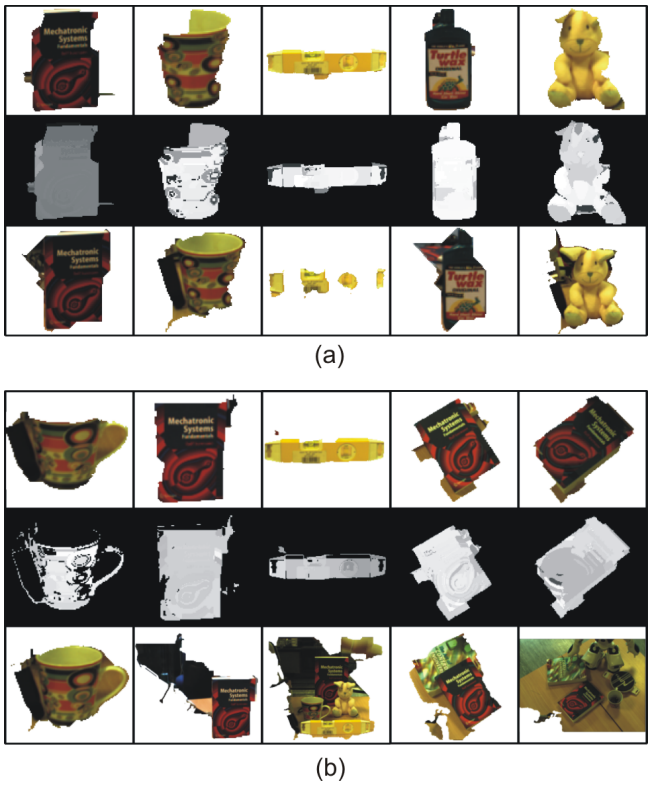

Figure 8: Different segmentations tested on static (a) and changing (b) background. The top row are the results for M-L, the centre shows the probability map the bottom row are results of the method described by Fitzpatrick.

The result from GraphCuts is shown in this image, (Fitzpatrick, 2003) goes on to extract the connected region which is closest to the end-effector when the object is touched for the first time, further degrading the quality of the final segmentation. Problems like these contribute to the wide standard deviation shown in Figure 7(b).

The changing backgrounds have a significant effect on the subtraction used by (Fitzpatrick, 2003) (bottom row), and in some cases results in the whole image being segmented as part of the object as in Figure 8(b). Our results show a large improvement (top row). (Fitzpatrick, 2003) failed for a number of different reasons, the right hand column of Figure 8(b) shows a frame from video with camera shake. In the fourth column, movement of the book causes the object behind to move slightly. Columns two and three include people moving behind the robot. Any level of movement or change in the environment causes the background subtraction used in Fitzpatrick to fail.

6. Results - 3D Arm Motions

Extra datasets are used to test the method on objects undergoing challenging motions. 10 videos of the robot pushing a bottle on a table and undergoing significant rotations are used to assess the performance of the algorithm under non-linear motions. Two further datasets (10 videos each) containing a box and spirit level being gripped and moved in space by the

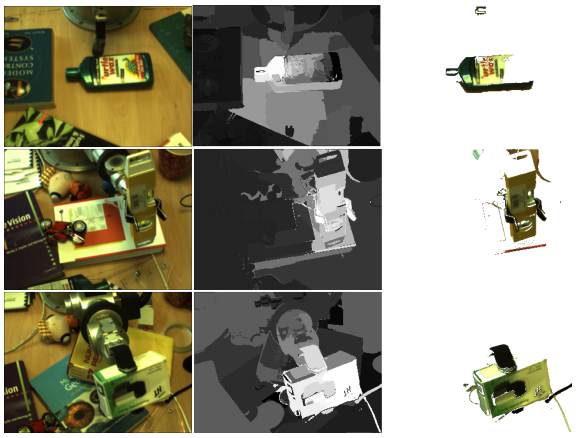

Figure 9: Segmentations and probability maps computed using a homography motion model.

robot are used to test the model under 3D motions. In this experiments, both the translation and homography transformation methods are compared. The results are compared to a hand segmented ground truth for a random frame in the video sequence.

The qualitative performance of the algorithm using a homography motion model is shown in Figure 9. The columns show a frame of the video, the probability map and resulting segmentation respectively. The first row shows a segmented frame of the video where the object undergoes significant rotation during the push. The second and third rows are videos of an object (spirit level and box) being held by the robot and moved non-linearly in front of the camera.

Figure 10 shows the Precision-Recall graphs for a homography model and the translation model. The graphs show a similar performance for both models for the pushed bottle (Figure 10(a)). This is due to (i) inaccuracies in calculating the homography for small movements in a segment, and (ii) the good approximation of translations only for small motions.

Figures 10(b) and 10(c) show results for objects which are held by the robot and moved non-linearly and in 3D in front of the cameras. Here, the homography based method shows a significant improvement over the translation model.

The spirit level tested is accurate near the manipulator, but becomes less accurate towards the edges. This is due to the quality in estimating the homography further away from the manipulator. High velocity motion at the edge of the level is approximately linear in these regions and if it is calculated inaccurately will not match well with the arm motion. In Figure 10(b) a smaller object is used, with improved precision and recall. The linear model also works much better in this case but is still out performed by a homography.

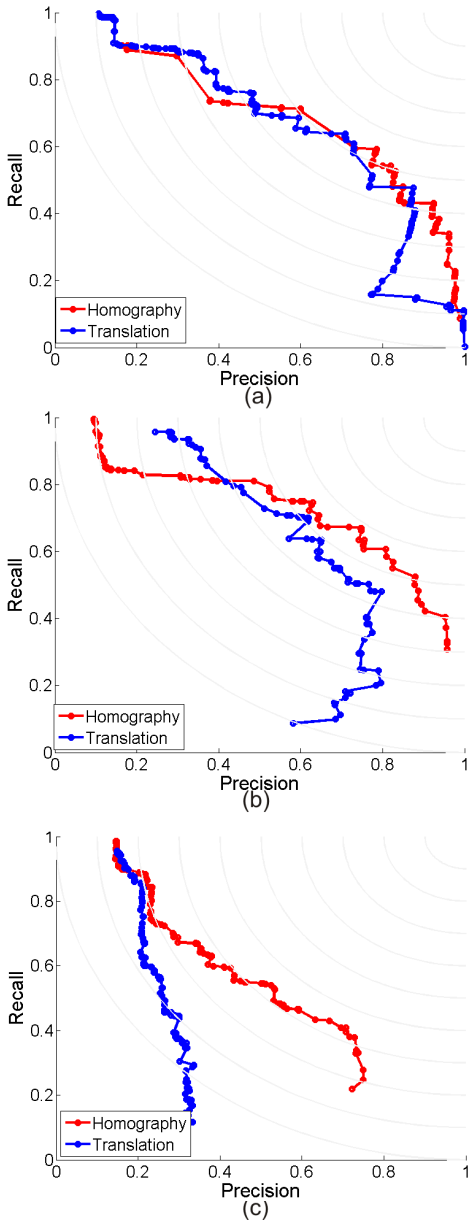

Figure 10: P-R graphs comparing Homography and Translation motion functions. (a) Rotating bottle, (b) held box and (c) held spirit level

7. Conclusion and Discussion

A new method for localising objects based on visual information and robot motion has been introduced. Alternative state of the art methods use image differencing as a basis for the localisation; despite its ease of use and ability to cope with cluttered, or unstructured backgrounds, it breaks down if the background environment is dynamic. Empirical results show that our method provides a much better quality of localisation compared to previous results, with both, dynamic and static backgrounds. To the best of our knowledge, this is the first implementation of motion-based localisation using a robot where the background can change and no prior knowledge is used. The results in this section show that our method clearly improves results, as seen most clearly in Figure 8.

The probabilistic formulation of the algorithm gives a measure of certainty in regions of the image belonging to the object. This will form a basis for the robot to decide where the best

place to manipulate the object from in future, further differentiating the object from it's background. The paper also illustrates how the method can be extended from poking objects in a linear fashion to cases in which the objects are moved in complex 3D trajectories. This is based on a homography motion model which assumes that points within segments lie in a plane. This assumption could be relaxed if full 3D data (i.e. stereo camera system) is used. In this case, a standard 3D rotation-translation matrix could be used as a motion model for rigid objects.

There are limits to our approach, so there is plenty of scope for future work. For example, we inherit any problems in the low-level segmentation algorithm. The low-level algorithm can easily be replaced to suit the particular application or environment, and note segmentation is a major and active field within Computer Vision; we can take advantage of new developments.

A problem that is more direct is the class of objects we can localise. Our current motion models assume rigid bodied objects. Flexible and articulated objects would require complex non-linear motion models for their segmentation. In principle, the method introduced here could be recursively used for each connected component of an articulated body.

Practical problems exist too. In section 4.2 the complexity of the algorithm is mentioned. There are K frames and L unique segments in the video, and the mean flow vector $\mathbf{b}_S^i$ needs to be calculated for each, giving KL motion models. Unless the distribution is updated for each frame as the video stream is being processed, the complexity will rise linearly with the number of frames. Using this method for large sections of video could become computationally difficult. To ensure the algorithm is practical for robotic applications it can be modified to work incrementally. Section 4.5 shows how the distributions in multiple frames are integrated in to one. This process is equivalent to building the distributions at each frame using a running average.

It may also be the case some frames are very noisy, due to high-frequency camera shake the the head moves for example. If we have some measure which gives a confidence in particular frames of the video, the robot may have some prior knowledge about any noise in the images which affect its vision negatively. This can be formulated as a prior probability on the frames, and incorporated into Equation 16, further increasing the stability and robustness of the final object segmentation. Again, this is not considered in the paper, but using prior knowledge about the robots sensors and objects in the scene is something that will be incorporated into future work.

Nonetheless, this paper contributes a vision/touch based localisation algorithm that is shown to be robust to moving background clutter and which could be the basis for further applications, including motion planning and recognition.

References

P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. *From Contours to Regions: An Empirical Evaluation* International Conference on Computer Vision and Pattern Recognition CVPR 2009.

A. Arsenio, P. Fitzpatrick, C. C. Kemp, G. Metta, and I. Genova *The whole world in your hand: Active and interactive segmentation*, Proceedings of the Third International Workshop on Epigenetic Robotics, 49-56, 2003.

J. Y. Bouguet, *Pyramidal implementation of the lucas kanade feature tracker description of the algorithm* Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.

Y. Boykov and V. Kolmogorov. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(9), 1124-1137, 2004.

M. Boshra and H. Zhang, *Localizing a polyhedral object in a robot hand by integrating visual and tactile data*, International Journal of Pattern Recognition , 33(3) 483-501, 2000.

T. Brox and J. Malik, *Object Segmentation by Long Term Analysis of Point Trajectories*, Lecture Notes in Computer Science, European Conference on Computer Vision, 282-295, 2010.

A. Blake and J.M. Brady, *Computational Modelling of Hand-eye corrdination* Philosophical Transactions: Biological Sciences, 337(1281), 351-360, 1992.

F. Chaumette and S. Hutchinson *Visual servo control. I. Basic approaches* IEEE Robotics & Automation Magazine, 13(4), 82-90, 2006.

F. Chaumette and S. Hutchinson *Visual servo control. II. Advanced approaches* IEEE Robotics & Automation Magazine, 14(1), 109-118, 2007.

Y. Cheng *Mean shift, mode seeking, and clustering* IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8), 790-799, 1995.

D. Comaniciu and P. Meer, *Mean shift: A robust approach toward feature space analysis*,IEEE Transactions on pattern analysis and machine intelligence, 24(5), 603, 2002.

P. Espinace, D. Langdon, A. Soto *Unsupervised identification of useful visual landmarks using multiple segmentations and top-down feedback* Robotics and Autonomous Systems, 56(6), 538-548, 2008.

P. F. Felzenszwalb and D.P. Huttenlocher, *Efficient graph-based image segmentation*, International Journal of Computer Vision, 59(2), 167-181, 2004.

P. Fitzpatrick, *First contact: an active vision approach to segmentation*, In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3, 2003

P. Fitzpatrick and G. Metta *Grounding vision through experimental manipulation* Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 361(1811), 2165, 2003.

R. Hartley and A. Zisserman *Multiple View Geometry - Second Edition* Cambridge University Press, 2000.

Y. Huang, Q Liu, and D. Metaxas, *Video Object Segmentation by Hypergraph Cut*, International Conference on Computer Vision and Pattern Recognition, 2009.

J. Kenney, T. Buckley, O. Brock *Interactive Segmentation for Manipulation in Unstructured Environments* IEEE International Conference on Robotics and Automation, 2009.

D. G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, International Journal of Computer Vision, 60(2), 91-110, 2004.

D. R. Martin, C. C. Fowlkes, and J. Malik, *Learning to detect natural image boundaries using local brightness, color, and texture cues* IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(5), 2004.

G. Metta, and P. Fitzpatrick *Early integration of vision and manipulation* Adaptive Behavior, 11(2), 109-128, 2003.

R.A. Newcombe and A.J. Davison, *Live dense reconstruction with a single moving camera*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1498-1505, 2010.

S. Paris *Edge-preserving Smoothing and Mean-shift Segmentation of Video Streams* Processing of the European Conference on Computer Vision (ECCV), 460-473, 2008.

O.A. Pellejero, C. Sagüés, and J.J. Guerrero, , *Automatic Computation of the Fundamental Matrix from Matched Lines*, Current Topics in Artificial Intelligence, 197-206, 2004.

M. G. Ross *Exploiting Texture-Motion Duality in Optical Flow and Image Segmentation* Masters thesis, Massachusetts Institute of Technology.

P. H. Schnemann *A generalized solution of the orthogonal procrustes problem* Psychometrika 31(1), 1-10, 1966.

M. W. Spong, and M. Vidyasagar, *Robot Dynamics and Control*, Wiley-India, 2009.

A. Stein, D. Hoiem, and M. Hebert, *Learning to find object boundaries using motion cues* International Conference on Computer Vision, ICCV 2007.

P. H. S. Torr and A. Zisserman *Concerning Bayesian Motion Segmentation, Model Averaging, Matching and the Trifocal Tensor* European Conference on Computer Vision, 511-527, 1998.

P. H. S. Torr. *Geometric Motion Segmentation and Model Selection*. Philosophical Transactions of the Royal Society, 356, 1321-1340, 1998.

A. Ude, D. Omrcen, G. Cheng, *Making Object Learning and Recognition an Active Process* International Journal of HUmanoid Robotics, 5(2), 267-286, 2008.

A. Vazquez-Reina and S. Avidan and H. R. Pfister and E. Miller, *Multiple Hypothesis Video Segmentation from Superpixel Flows*, Lecture Notes in Computer Science, European Conference on Computer Vision, 268-281, 2010.

L. E. Weiss, A. C. Sanderson, C. P. Neuman, *Dynamic Sensor-Based Control of Robots with Visual Feedback*, IEEE Journal on Robotics and Automation, 3(5), 404-417, 1987.

K. Welke, J. Issac, D. Schiebener, T. Asfour, R. Dillman *Autonomous Acquisition of Visual Multi-View Object Representations for Object Recognition on a Humanoid Robot* IEEE International Conference on Robotics and Automation, 2012-2019, 2010.

R. Bryan Williamson, *Interactive Perception for Cluttered Environments*, PhD Thesis, Clemson University, 2009.

Z. Yin and R. Collins *Shape Constrained Figure-Ground Segmentation and Tracking* International Conference on Computer Vision and Pattern Recognition, 2009.

Daniel Beale received his M.Math. degree in Mathematics from the University of Bath in 2008. He is currently a Ph.D. Candidate in the Department of Computer Science at the University of Bath. His research interests are in the areas of computer vision, pattern recognition, and robotic systems with emphasis on the probabilistic fusion of robotic motion and vision.

Pejman Iravani is an RCUK Research Fellow in the Department of Mechanical Engineering at the University of Bath. Prior to this, he gained his PhD from The Open University with a thesis in the area of machine learning for multi-robot control. He his currently working on the areas of model learning for compliant robot actuation and object modelling and recognition using machine vision.

Peter Hall is a Reader (tenured Associate Professor) in Computer Science at the University of Bath. His research interests include computer vision, computer graphics, and machine learning. His PhD is in scientific visualisation. He is an executive member of the British Machine Vision Association.